Object shape estimation and modeling, based on sparse Gaussian process implicit surfaces, combining visual data and tactile exploration

Gandler, Gabriela Zarzar; Ek, Carl Henrik; Björkman, Mårten; Stolkin, Rustam; Bekiroglu, Yasemin

DOI:
10.1016/j.robot.2020.103433

Document Version
Publisher's PDF, also known as Version of record

Citation for published version (Harvard):
Gandler, GZ, Ek, CH, Björkman, M, Stolkin, R & Bekiroglu, Y 2020, 'Object shape estimation and modeling, based on sparse Gaussian process implicit surfaces, combining visual data and tactile exploration', *Robotics and Autonomous Systems*, vol. 126, 103433, pp. 1-16. https://doi.org/10.1016/j.robot.2020.103433

Robotics and Autonomous Systems 126 (2020) 103433

Contents lists available at ScienceDirect

Robotics and Autonomous Systems

journal homepage: www.elsevier.com/locate/robot

Object shape estimation and modeling, based on sparse Gaussian process implicit surfaces, combining visual data and tactile exploration

Gabriela Zarzar Gandler [a,*,1], Carl Henrik Ek [b], Mårten Björkman [c], Rustam Stolkin [d], Yasemin Bekiroglu [d,1]

[a] Peltarion, Holländargatan 17, 11160 Stockholm, Sweden
[b] University of Bristol, Beacon House, Queens Road, Bristol BS8 1QU, UK
[c] KTH Royal Institute of Technology, Brinellvägen 8, 11428 Stockholm, Sweden
[d] University of Birmingham, Edgbaston, B15 2TT, Birmingham, UK

ARTICLE INFO

Article history:
Received 31 May 2019
Received in revised form 2 December 2019
Accepted 7 January 2020
Available online 15 January 2020

Keywords:
Tactile sensing
Shape modeling
Implicit surface
3D reconstruction
Gaussian process
Regression

ABSTRACT

Inferring and representing three-dimensional shapes is an important part of robotic perception. However, it is challenging to build accurate models of novel objects based on real sensory data, because observed data is typically incomplete and noisy. Furthermore, imperfect sensory data suggests that uncertainty about shapes should be explicitly modeled during shape estimation. Such uncertainty models can usefully enable exploratory action planning for maximum information gain and efficient use of data. This paper presents a probabilistic approach for acquiring object models, based on visual and tactile data. We study Gaussian Process Implicit Surface (GPIS) representation. GPIS enables a non-parametric probabilistic reconstruction of object surfaces from 3D data points, while also providing a principled approach to encode the uncertainty associated with each region of the reconstruction. We investigate different configurations for GPIS, and interpret an object surface as the level-set of an underlying sparse GP. Experiments are performed on both synthetic data, and also real data sets obtained from two different robots physically interacting with objects. We evaluate performance by assessing how close the reconstructed surfaces are to ground-truth object models. We also evaluate how well objects from different categories are clustered, based on the reconstructed surface shapes. Results show that sparse GPs enable a reliable approximation to the full GP solution, and the proposed method yields adequate surface representations to distinguish objects. Additionally the presented approach is shown to provide computational efficiency, and also efficient use of the robot's exploratory actions.

1. Introduction

Humans and animals sense environments via multiple sensory modalities. It has been shown that both visual and haptic modalities provide strong contributions to human perception of objects [1–3]. Furthermore, it has been shown that visual and haptic 3D shape information are integrated by humans in a statistically optimal way, and bimodal perceptions of 3D shapes are more accurate than those based on either vision or touch alone [3]. Inspired by these findings, we study how robots can complement visual information with tactile sensing, to achieve more robust perception of 3D object shapes. Since camera views and tactile samples typically provide incomplete and noisy information, we believe that it is important that inferred 3D shape models also encode the uncertainties associated with these shape estimates. We therefore propose a method to acquire 3D object models probabilistically, in a computationally efficient way. We demonstrate our method with empirical experiments, with data obtained from two different robotic manipulator platforms when interacting with a variety of different benchmark object types and shapes, using visual and tactile sensors.

Acquiring object shape information is important in numerous robotics applications, e.g. for grasp and motion planning [4,5], which in turn may rely on estimating an object's pose [6]. However, in autonomous service-robot scenarios where novel objects are encountered, 3D object models are often unavailable a priori. Various approaches have been proposed for estimating object shapes. One approach [7] is to rely on the assumption that many objects (especially in structured, man-made environments) have symmetries. Assumptions of symmetry can be used

* Corresponding author.
E-mail address: gabriela@peltarion.com (G. Zarzar Gandler).

[1] The authors were with ABB Corporate Research, Sweden and G. Zarzar Gandler was also with KTH Royal Institute of Technology, Sweden. She is currently with Peltarion.

https://doi.org/10.1016/j.robot.2020.103433

to complete 3D models from partial visual observations, by finding the best symmetry plane and reflecting the observed points accordingly. However symmetry assumptions do not hold for many objects, e.g. the handle on a mug, or rubble in a disaster-response scenario, or robotic handling of hazardous waste [8–10]. Alternatively, instead of relying on data from one view only, surface reconstruction can be performed by exploring the object from various view directions [11]. For example, a robot may move an object in front of a depth camera (or move its camera relative to stationary object), prioritizing surface regions that maximize the information gain while minimizing the movement cost. Such systems can be improved by incorporating an additional sensory modality. For example, tactile samples, from an occluded region of an object, can be used to augment a partial point-cloud view of the surface nearest to a depth camera. This way, an initial hypothesis based on a symmetry assumption can be refined after grasping the object, given tactile and proprioceptive data from fingers which contact the object. Model refinement can then become iterative, e.g. using an extended Kalman filter approach [12].

Extracting shape information is challenging mainly due to the fact that sensory data can be incomplete and noisy. Furthermore, a key aspect in modeling shape information is the surface representation that allows for efficient action selection. In this context, Gaussian process (GP) is a promising modeling framework to encode information about object shape [13,14]. It provides parametrization of uncertainty, which adds crucial descriptiveness to the representation of object shapes. This is especially useful for tasks such as robotic grasping, facilitating more informed planning to have a good trade-off between exploration and exploitation, i.e. choosing more certain object areas that would lead to successful grasp configurations or choosing where to explore the object in less certain areas. Therefore GPs have been shown to be effective for various types of applications such as guiding motions for exploring objects, reconstructing surfaces and mapping incomplete and occluded regions [13,15,16]. For these applications, a surface representation based on GPs, Gaussian Process Implicit Surface (GPIS), has been used, as uncertainty in GP formulation enables making decisions about where to explore next. Many recent works use the standard full GPIS formulation for surface representation, e.g. through an active touch strategy to reduce surface geometry uncertainty by using one robot finger equipped with a tactile sensor [17], focusing on sliding paths instead of touch locations [18], or following a classification approach to guide the exploration [19]. Hence, explorative action selection can be performed by considering the trade-off between the uncertainty in the surface estimation and the corresponding travel cost [11,20]. However usually these works either rely on unimodal data (visual or tactile) or suffer from computational complexity of full GP formulation. In this paper, we present a computationally efficient GPIS formulation applied to real-world bimodal sensory data (visual and tactile) for surface modeling.

In our previous work, [15] we built GPIS models based on actively exploring objects, using visual and tactile sensing, where a surface is induced by the level-set of a continuous function estimated via GP regression. The proposed framework focused on identifying most uncertain surface areas to prioritize for tactile exploration. After building first surface hypothesis via visual data, the surface estimation was further improved by gradually adding tactile information, refining object models initially learned from visual points. This paper presents a number of extensions of the previous work. Differently from the aforementioned approaches we focus on computationally efficient probabilistic representations of real sensory data from visual and tactile observations of objects.

Our formulation builds on the GPIS representation. GPs allow to combine prior beliefs about object shape properties with knowledge from observations in a principled manner while modeling inherent uncertainties. The GP prior is controlled by the choice of covariance (or kernel) function, which provides structure incorporating our prior knowledge of the shape of the surface, e.g. thin-plate covariance function [15,21]. Another approach is to use shape primitives to define geometric object priors [22], or non-stationary kernels for reconstruction of 3D surfaces [23]. We study how different kernel functions affect object surface representation. Kernels are parameterized by hyperparameters [24] which determine the prior knowledge encoded in the kernel. Different methods have been proposed to find hyperparameters, e.g. maximization of the marginal likelihood [24]. However this becomes slower to compute as the size of the solution space grows, commonly creating a trade-off between modeling uncertainty and computational speed. To circumvent the computational complexity associated with learning using GPs, we optimize a variational lower bound on the marginal likelihood using the approach presented in [25]. This alternative sparse GP formulation has a much lower computational demand and allows surface modeling even when raw data is relatively large.

Main contributions and differences to our previous work [15] can be summarized as follows:

- We use sparse Gaussian processes for surface reconstruction, which results in approximate solutions with less computational time in comparison to standard GPIS, while preserving accuracy.
- We investigate different configurations for GPIS, e.g. different kernel choices, and optimize model parameters through variational learning, rather than choosing manually as in [15], and present extensive evaluations and analysis of the results. For evaluating reconstruction quality, we apply i.a. a clustering method to assess how well the reconstructions group into semantically meaningful classes of objects.
- The method is evaluated both using synthetic (perfectly smooth and generally complete surfaces) and real data from two different robotic platforms.

2. Methodology

This section briefly introduces our approach for modeling and understanding shape using probabilistic representations. We present the description of the outline of the system used to extract shape information from raw data, as well as the data sets used for evaluating the system.

2.1. System outline

The proposed system outline is illustrated in Fig. 1. There are two main steps in the system: surface reconstruction from 3D data points obtained from visual and haptic sensors and clustering of these surface models. The visual and haptic measurements are pre-processed by centering, normalizing and finally enhancing the data with topological information, in the form of points added inside and outside object point clouds. We use GP as a Bayesian prior, which expresses the beliefs about the latent function we aim to model, before any data is taken into account. Hence we select a kernel function to form the covariance and induce the smoothness of the GP prior. It follows that parameters are optimized [25] by means of maximizing a variational lower bound to the exact log marginal likelihood, using the L-BFGS-B [26,27] algorithm. The sparse GP posterior distribution is further derived, whose mean serves as basis to extract a level-set, using the Marching Cubes algorithm [28]. The level-set in its turn induces the object implicit surface, which is represented by

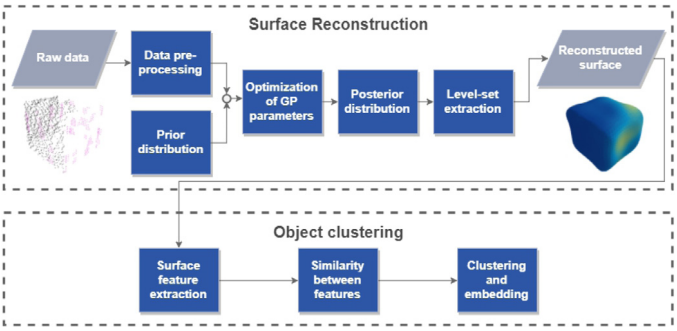
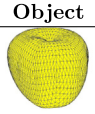


Fig. 1. The system outline for sparse Gaussian process implicit surfaces for object shape modeling.

Table 1
The Princeton ModelNet objects from four different categories: apples, bottles, pots and ducks.

Name	Object	# of points	Name	Object	# of points
apple1		1084	pot1		2659
apple2		832	pot2		6879
apple3		3575	pot3		1843
bottle1		2287	duck1		2112
bottle2		6576	duck2		2329
bottle3		5228	duck3		4846

a triangular mesh. For further theoretical details about GPs and kernel functions refer to Section 3.

In order to tackle object clustering, firstly local features are extracted from the mesh. We particularly exploit principal curvatures,[2] [29,30], which are representations that relate to the grasping actions the object can afford [15]. Later distances are measured through the kernel two-sample test [31], based on the objects' local feature representations. These distances are further translated into similarity measures through a Gaussian kernel.[3] Finally spectral clustering and embedding [32] are undertaken to discriminate objects, based on the similarity measures.

2.2. Data sets

Our study employs data from different sources, as a means to collect various insights about 3D shape modeling. With this in mind, two essential data types are evaluated:

- A synthetic public data set, called Princeton ModelNet data set [33], with object point clouds from different object categories and
- A real sensory data set, which includes samples from [15], as well as samples acquired from a PR2 robot. The object point clouds contain measurements from both visual and tactile sensors.

The Princeton ModelNet data set [33] contains a clean collection of 3D CAD models for objects in various categories, which are publicly available. The corresponding labels were obtained by Princeton researchers via Amazon Mechanical Turk service and are provided freely. Objects from four categories are used in the experiments: apples, bottles, pots and ducks. We choose example categories that range from simple to more complicated in shape, in order to demonstrate how our method performs on objects with different surface characteristics, curvatures and scales of shape variation. These objects range from comparatively regular shapes, such as spherical and cylindrical objects as in apples and bottles, to more irregular shapes with many concavities and sharp edges, e.g. pots and ducks.

The raw data for each object is a point cloud, i.e. a collection of positions in the 3D space. Table 1 displays, for each object, the corresponding name, CAD model and number of data points. The object names are used throughout the rest of this work as a means to refer to each object. As one can see in Table 1, the number of points in the point clouds varies considerably: the most scarce cloud has 832 points, while the densest has 6879 points. The figures of the CAD models in Table 1 are obtained from [34].

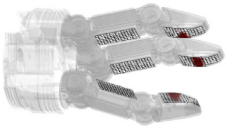
[2] We follow the method in [29] but adapt the estimation of normal vectors as in [30].

[3] The similarity measure $w^{i,j}$ between object i and j is calculated as $w^{i,j} = exp\left(\frac{-(d^{i,j})^2}{2\sigma_w^2}\right)$, where $d^{i,j}$ and $\sigma_w = 0.05$ stand for the distance measure and the width of the Gaussian kernel, respectively.

Table 2
Details about the objects utilized in the experiments using the robot with KUKA arm and Schunk hand [15] (left) and a PR2 robot (right).

Name	Object	# of sensory points *Visual*	# of sensory points *Tactile*	# of ground-truth points	Name	Object	# of sensory points *Visual*	# of sensory points *Tactile*
box1		6979	1714	122413	box1		1919	190
box2		5450	445	106497	box2		3113	207
box3		12620	1952	209962	box3		3911	524
cyl1		5029	1580	103662	cyl1		3290	323
cyl2		4528	1460	101707	cyl2		5465	676
cyl3		2765	829	60923	cyl3		3948	620
cyl4		5071	1975	114010	cyl4		4431	737
spray1		4252	1214	106690	bottle1		3759	478
spray2		4084	1508	104193	bottle2		3044	327
spray3		2937	1166	71734	bottle3		2915	321

(a)

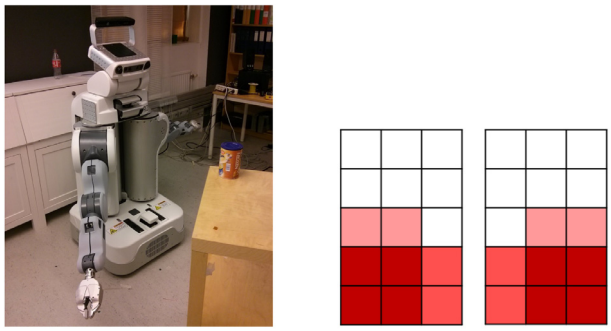


(b)

Fig. 2. The experimental robot platform from [15]. (a) KUKA arm with a Schunk hand is located on the left, while Kinect camera is on the right (modified from [15]). (b) The Schunk hand in detail, with the thumb opposing the other two fingers. There are two tactile pads in each finger with 14 × 6 and 13 × 6 cells. The red regions correspond to tactile readings from the sensor cells where hypothetical contacts are sensed. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

The real sensory data utilized in the experiments, in Table 2, includes the data used in [15]. The data was collected using a setup composed of a fixed Kinect stereo vision camera and an industrial KUKA arm (6 degrees of freedom) with a Schunk hand (7 degrees of freedom), which was equipped with three fingers. The fingers had pressure sensitive tactile pads and the object to be explored was placed on a table, as in Fig. 2a. Fig. 2b shows example tactile readings from the Schunk hand. During exploration the fingers were closed until contacts were sensed. More detailed information regarding the experimental robot platform can be found in [15].

The data itself consists of a collection of 3D point coordinates, i.e. a point cloud. Data collection was performed through a perceptually-guided procedure, which is explained below. For each object, firstly a point cloud was recorded from the visual sensor. Then the robotic hand was guided to touch the objects at different locations to gather tactile observations. The goal was to progressively refine the surface reconstructions by guiding the hand toward surface points for which the predictive variance was large, in order to explore the most uncertain surface regions and decrease uncertainty. This procedure considered an action space defined by 6 different heights and 9 different approaching angles. Through this procedure at most 54 tactile readings were recorded, complementing the original visual data. For some objects fewer touches were applied, due to their lower heights. Data from both sensors were defined in the same frame, thanks to the calibration performed on the arm–hand configuration with respect to the camera system, with a precision of few millimeters [15].



(a) (b)

Fig. 3. The experimental PR2 robot platform. (a) The PR2 robot and (b) an illustration of tactile pads containing 5 × 3 cells on each finger.

The objects belong to three different basic shape categories, namely boxes, cylinders and spray bottles. The objects from these categories were chosen as they were of varying shapes and large enough to be compatible with the robotic hand, so that the objects were not too small or large for the hand to acquire reasonable tactile data. The Schunk hand has relatively large tactile pads, a 13 × 6 array on each distal link and a 14 × 6 array on each proximal link, with each tactile cell corresponding to a spatial resolution of 3.5 mm. Due to the size and spatial resolution of the tactile pads and the preshape pose of the hand (Fig. 2b), tactile readings (3D points) may fail to capture fine details on a surface, e.g. small concavities which are of finer scale than the relatively coarse taxels.

As one can see in Table 2, the number of points in the sensory point clouds varies considerably: taking into consideration all visual and tactile readings, the most scarce cloud has 3594 points, while the densest has 14 572 points. The ground-truth point clouds for each object are derived from high resolution scans using a turntable setup, and are denser than point clouds captured by the robots' lower resolution depth cameras. Since the ground-truth scans from a turntable often have missing points in the top and bottom surfaces of objects, additional points are added in those areas to correct and complete the ground-truth models.

The real sensory data in Table 2 also includes visual and tactile observations acquired using a PR2 robot, shown in Fig. 3a. The robot hand – equipped with two fingers and tactile pads such as in Fig. 3b – was guided to touch the objects at different locations to gather tactile observations. The action space was defined by 9 different heights (with a spacing of 2 cm) and 7 different approaching angles (approaching objects from angles between −60 degrees and +60 degrees with a spacing of 20 degrees). Thus at most 63 tactile readings were recorded, complementing the original visual data. Note once more that for some objects fewer touches were applied. Differently from the Princeton ModelNet data set, the real sensory data set contains considerably noisy measurements, both from the visual and the tactile sensors.

3. Sparse Gaussian process implicit surfaces

3.1. Preliminaries

The problem tackled by this work is, in essence, a regression problem. Consider the function $f(\mathbf{x})$, where $\mathbf{x} = [x_1, x_2, x_3]^T$ represents an input location and $f : \mathbb{R}^3 \to \mathbb{R}$. The mapping f is unknown – it is a latent function – and all one can observe is a data set $S = \{(\mathbf{x}_i, y_i) | i = 1, \ldots, n\}$ with input locations $\mathbf{x}_i \in \mathbb{R}^3$ and corresponding noisy observations $y_i \in \mathbb{R}$. We define $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]^T \in \mathbb{R}^{n,3}$ and $\mathbf{y} = [y_1, \ldots, y_n]^T \in \mathbb{R}^n$. The relation between function values and corresponding noisy observations is considered to be:

$$y_i = f(\mathbf{x}_i) + \epsilon_i, \tag{1}$$

where $\epsilon_i \sim \mathcal{N}(0, \sigma_n^2)$ are noise terms that are independent and identically distributed. They are assumed to follow a zero-mean Gaussian distribution with variance σ_n^2.

A GP is a stochastic process, i.e. a collection of (infinitely) many random variables, any finite number of which is jointly Gaussian distributed [24]. A GP can also be interpreted as a distribution over functions $f(\mathbf{x})$,

$$f(\mathbf{x}) \sim \mathcal{GP}(0, k(\mathbf{x}, \mathbf{x}')), \tag{2}$$

where $\mathbf{x}, \mathbf{x}' \in \mathbb{R}^3$, the GP mean is zero and the GP covariance is $k(\mathbf{x}, \mathbf{x}')$, i.e. a kernel function $k : (\mathbb{R}^3, \mathbb{R}^3) \to \mathbb{R}$.

Consider $\mathbf{x}^*$ to represent some new input test point outside the set S and $\mathbf{X}^* \in \mathbb{R}^{n^*,3}$ a concatenation of n^* such test points. The distribution of $\mathbf{f}^* = [f(\mathbf{x}_1^*), \ldots, f(\mathbf{x}_{n^*}^*)]^T$ can be derived from the joint distribution of $\mathbf{y}$ and $\mathbf{f}^*$, and is given by

$$\begin{aligned} &p(\mathbf{f}^* | \mathbf{y}, \mathbf{X}, \mathbf{X}^*) \sim \mathcal{N}(\boldsymbol{\mu}^*, \boldsymbol{\Sigma}^*), \text{ where} \\ &\boldsymbol{\mu}^* \triangleq \mathbb{E}[\mathbf{f}^* | \mathbf{y}, \mathbf{X}, \mathbf{X}^*] \\ &\quad = K(\mathbf{X}^*, \mathbf{X})[K(\mathbf{X}, \mathbf{X}) + \sigma_n^2 I]^{-1}\mathbf{y}, \\ &\boldsymbol{\Sigma}^* = K(\mathbf{X}^*, \mathbf{X}^*) \\ &\quad - K(\mathbf{X}^*, \mathbf{X})[K(\mathbf{X}, \mathbf{X}) + \sigma_n^2 I]^{-1} K(\mathbf{X}, \mathbf{X}^*), \end{aligned} \tag{3}$$

where $K(\mathbf{X}, \mathbf{X}) \in \mathbb{R}^{n,n}$, $K(\mathbf{X}^*, \mathbf{X}^*) \in \mathbb{R}^{n^*,n^*}$ and $K(\mathbf{X}, \mathbf{X}^*) \in \mathbb{R}^{n,n^*}$ are filled with elements $k(\mathbf{x}_i, \mathbf{x}_j)$, $k(\mathbf{x}_i^*, \mathbf{x}_j^*)$ and $k(\mathbf{x}_i, \mathbf{x}_j^*)$ respectively, for i denoting a row index and j, a column index. Thus $K(\mathbf{X}^*, \mathbf{X}) = K(\mathbf{X}, \mathbf{X}^*)^T$.

3.2. Gaussian process implicit surface

GP regression is applied to estimate a continuous latent function f, which in this context should reveal, for each point in the 3D space, whether it is part of the object or not. In order to achieve this, we need to define how f should vary in the 3D domain, i.e.: $f(\mathbf{x}_i) = 0$ for $\mathbf{x}_i$ on the object surface, $f(\mathbf{x}_i) > 0$ for $\mathbf{x}_i$ outside the surface and $f(\mathbf{x}_i) < 0$ for $\mathbf{x}_i$ inside the surface. Therefore, through this topological constraint, the zero-valued level of f induces an implicit surface (f is set to zero without loss of generality). This is materialized by adding topological information to the system, i.e. the set S is enhanced with added points both inside and outside the object: points inside and outside the object are mapped to $y = -1$ and $y = 1$, respectively. As the implicit surface is induced by the zero-valued level of f, input sensory observations are mapped to $y = 0$. Since f is represented by a GP, the implicit object surface is called Gaussian Process Implicit Surface (GPIS) [13], which allows to fuse uncertain data from sensors in a probabilistic shape estimate.

3.3. Variational sparse Gaussian process

Probably the most significant limitation of GPs is how its computational demand scales with n (the data size): the associated complexity is $\mathcal{O}(n^3)$ [24], which is prohibitive in terms of computational time and storage for large data sets. Therefore sparse approximate methods have been developed in the literature [24,25,35,36]. Most of these methods are associated with a computational cost of $\mathcal{O}(n(n^u)^2)$, for $n^u \ll n$. They adopt n^u pairs of auxiliary input–output variables, which are called inducing variables, denoted as $\mathbf{x}_i^u \in \mathbb{R}^3$ and $f_i^u \in \mathbb{R}$, for $i = 1, \ldots, n^u$.

Approximate sparse approaches that fit into the framework presented by [35] explicitly replace the covariance matrix $K(\mathbf{X}, \mathbf{X})$

using a low-rank approximation, which is ultimately equivalent to modifying the GP prior. However this is not an optimal way to approximate the exact GP, because it does not minimize any distance between the exact GP and its approximate counterpart [25]. Differently we employ an explicit approximation with respect to the exact GP posterior (in Eq. (3)), referred as $q(\mathbf{f}^*|\mathbf{X}^*)$.

Consider $q(\mathbf{f}^u) \sim \mathcal{N}(\boldsymbol{\mu}^{q(\mathbf{f}^u)}, \boldsymbol{\Sigma}^{q(\mathbf{f}^u)})$ to be a free variational Gaussian distribution, which will in general be different from the true posterior counterpart, i.e. $q(\mathbf{f}^u) \neq p(\mathbf{f}^u|\mathbf{y})$, where $\mathbf{f}^u = [f(\mathbf{x}_1^u), \ldots, f(\mathbf{x}_{n^u}^u)]^T$. The posterior distribution $q(\mathbf{f}^*|\mathbf{X}^*)$, that approximates the distribution in Eq. (3), is given by [25]

$$\begin{aligned} &q(\mathbf{f}^*|\mathbf{X}^*) \sim \mathcal{N}(\boldsymbol{\mu}^{q(\mathbf{f}^*|\mathbf{X}^*)}, \boldsymbol{\Sigma}^{q(\mathbf{f}^*|\mathbf{X}^*)}), \text{ where} \\ &\boldsymbol{\mu}^{q(\mathbf{f}^*|\mathbf{X}^*)} = K(\mathbf{X}^*, \mathbf{X}^u)K(\mathbf{X}^u, \mathbf{X}^u)^{-1}\boldsymbol{\mu}^{q(\mathbf{f}^u)}, \\ &\boldsymbol{\Sigma}^{q(\mathbf{f}^*|\mathbf{X}^*)} = K(\mathbf{X}^*, \mathbf{X}^*) \\ &\quad - K(\mathbf{X}^*, \mathbf{X}^u)K(\mathbf{X}^u, \mathbf{X}^u)^{-1}K(\mathbf{X}^u, \mathbf{X}^*) \\ &\quad + K(\mathbf{X}^*, \mathbf{X}^u)K(\mathbf{X}^u, \mathbf{X}^u)^{-1}\boldsymbol{\Sigma}^{q(\mathbf{f}^u)} \\ &\quad\quad K(\mathbf{X}^u, \mathbf{X}^u)^{-1}K(\mathbf{X}^u, \mathbf{X}^*), \end{aligned} \tag{4}$$

where $\mathbf{X}^u \in \mathbb{R}^{n^u,3}$ is a concatenation of all inducing inputs. Eq. (4) defines the general form of the sparse posterior GP, with complexity $\mathcal{O}(n(n^u)^2)$.

A principled procedure to determine the variational parameters – the parameters of the variational distribution $q(\mathbf{f}^u)$ and the inducing inputs $\mathbf{X}^u$ – is [25]: to minimize the Kullback–Leibler (KL) divergence from the exact joint posterior $p(\mathbf{f}, \mathbf{f}^u|\mathbf{y}, \mathbf{X})$ to the corresponding joint variational approximation $q(\mathbf{f}, \mathbf{f}^u|\mathbf{X})$, where $\mathbf{f} = [f(\mathbf{x}_1), \ldots, f(\mathbf{x}_n)]^T$. This is given by the following objective function:

$$\min KL(q(\mathbf{f}, \mathbf{f}^u|\mathbf{X}) \parallel p(\mathbf{f}, \mathbf{f}^u|\mathbf{y}, \mathbf{X})). \tag{5}$$

The minimization in Eq. (5) is equivalent to the maximization of a variational lower bound of the exact log marginal likelihood. Thus the lower bound can be maximized by analytically solving for the optimal choice of the parameters $\boldsymbol{\mu}^{q(\mathbf{f}^u)}$ and $\boldsymbol{\Sigma}^{q(\mathbf{f}^u)}$, given by [25]:

$$\begin{aligned} &q(\mathbf{f}^u) \sim \mathcal{N}(\boldsymbol{\mu}^{q(\mathbf{f}^u)}, \boldsymbol{\Sigma}^{q(\mathbf{f}^u)}), \text{ where} \\ &\boldsymbol{\mu}^{q(\mathbf{f}^u)} = \frac{1}{\sigma_n^2}K(\mathbf{X}^u, \mathbf{X}^u)\Big[K(\mathbf{X}^u, \mathbf{X}^u) \\ &\quad + \frac{1}{\sigma_n^2}K(\mathbf{X}^u, \mathbf{X})K(\mathbf{X}, \mathbf{X}^u)\Big]^{-1}K(\mathbf{X}^u, \mathbf{X})\mathbf{y}, \\ &\boldsymbol{\Sigma}^{q(\mathbf{f}^u)} = K(\mathbf{X}^u, \mathbf{X}^u)\Big[K(\mathbf{X}^u, \mathbf{X}^u) \\ &\quad + \frac{1}{\sigma_n^2}K(\mathbf{X}^u, \mathbf{X})K(\mathbf{X}, \mathbf{X}^u)\Big]^{-1}K(\mathbf{X}^u, \mathbf{X}^u). \end{aligned} \tag{6}$$

By optimizing the variational lower bound with respect to the inducing inputs $\mathbf{X}^u$, which are also considered free variational parameters, we find the optimal inducing inputs $\mathbf{X}^u$ for each object. The same is done for optimization of kernel hyperparameters and the variance of the Gaussian noise σ_n^2. Finally, the approximation $q(\mathbf{f}^*|\mathbf{X}^*)$ in Eq. (4) is employed with the fitted parameters from Eq. (6) as a proxy for the posterior, to perform inference.

3.4. Kernel functions

Kernel functions are key elements when solving regression problems through GPs, because they determine the properties of the functions considered for inference, such as smoothness and stationarity [24]. In this work 3 kernel functions are investigated: squared-exponential (SE), Matérn (MA) and thin-plate (TP) kernel functions.

The SE function is isotropic and gives rise to infinitely differentiable functions. It commonly has two hyperparameters: the variance (sometimes called intensity) σ^2 and the lengthscale l. It models well the smoothness characteristics of various random processes and can be expressed as

$$k^{SE}(\mathbf{x}, \mathbf{x}'|\boldsymbol{\theta}^{SE} = \{\sigma^2, l\}) = \sigma^2 \exp\left[-\frac{1}{2l^2}\|\mathbf{x} - \mathbf{x}'\|^2\right]. \tag{7}$$

The Matérn class (MA) of kernel functions is also isotropic, but induces functions with finite differentiability. It has the following hyperparameters: the variance σ^2, the lengthscale l and an additional parameter ν. As $\nu \to \infty$, the Matérn function converges to the SE. It includes not only the SE, but a large class of kernel functions, and proves very useful for applications due to this flexibility [37]. When $\nu = 5/2$, the Matérn function (MA52) is given by [24]:

$$\begin{aligned} &k^{MA52}(\mathbf{x}, \mathbf{x}'|\boldsymbol{\theta}^{MA52} = \{\sigma^2, l\}) \\ &= \sigma^2\left(1 + \sqrt{5}\frac{\|\mathbf{x} - \mathbf{x}'\|}{l} + \frac{5}{3}\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{l^2}\right) \\ &\quad \exp\left[-\sqrt{5}\frac{\|\mathbf{x} - \mathbf{x}'\|}{l}\right]. \end{aligned} \tag{8}$$

More recently the thin-plate (TP) kernel function has become a popular choice for GPIS estimation [15,21,22]. It is again an isotropic function and it has R as its only hyperparameter. It is set to $R = \max\|\mathbf{x} - \mathbf{x}'\|$, for $\mathbf{x}, \mathbf{x}' \in \mathbb{R}^3$. This means that R is the maximum Euclidean distance between input points. The TP kernel function can be expressed as

$$k^{TP}(\mathbf{x}, \mathbf{x}'|\boldsymbol{\theta}^{TP} = \{R\}) = 2\|\mathbf{x} - \mathbf{x}'\|^3 - 3R\|\mathbf{x} - \mathbf{x}'\|^2 + R^3. \tag{9}$$

Kernel functions can be combined in various different ways [38, p. 296]. Addition is one out of many possible operations to create new valid kernel functions. This work applies addition of kernel functions, which is described in Section 4.1.

4. Experiments

In this section we firstly present various aspects concerning the design choices of our solution and secondly explain how our experiments are divided.

4.1. Design choices

Data pre-processing. This involves centering, normalizing (scaling),[4] as well as enhancing the data with topological information, for every object. We start by addressing the data centralization. The raw data of synthetic objects is centered according to the mid-range center point, for every 3D coordinate. This performs well, because the synthetic data set virtually does not contain noise. The center of real sensory objects is however set to be the centroid of the visual points,[5] since a centroid is a safer measure of central location and therefore more suitable for noisy measurements. Finally, as described in Section 3.2, topological information needs to be incorporated, to enable the system discriminate the interior from the exterior of an object. This is done by adding to the original point cloud some points inside and outside the object, i.e. by augmenting the raw data with additional points. One internal point is added on the center, which is matched to the output value $y = -1$, while the external points are matched

[4] Only the data from synthetic objects is normalized, since these objects are originally in different scales.

[5] Visual points are the baseline observations for real sensory objects.

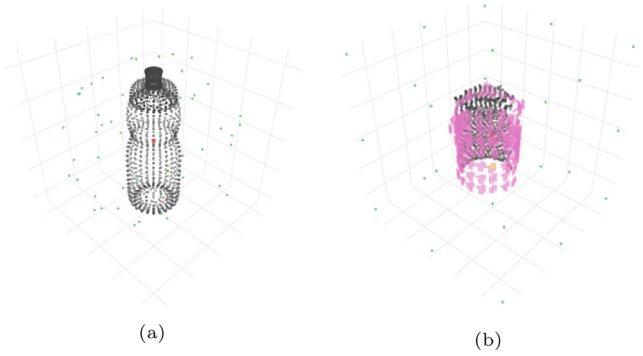


Fig. 4. A demonstration of how topological information is embedded in the model for objects from (a) the synthetic data set (bottle1) and (b) the real sensory data set from [15] (cyl1). Green points represent external points, while the red point is an internal point located on the center. In (b), black points stand for visual (for a better visualization, they were randomly sampled) and purple points for tactile measurements. One yellow point is added as an additional surface point, located on the bottom of the object and under the centroid, for all objects belonging to the real sensory data set. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

to $y = 1$. For objects belonging to the synthetic data set, 50 external points are randomly sampled from a centered sphere, whose ray is 10% larger than the distance between the center and the farthest surface point. For real sensory data, points are added on the borders of the object scene, which is characterized by the volume of a centralized cube with edge length 260 mm. Since these points are fixed (as apposed to randomly sampled), only 26 points are added. All objects are assumed to fit inside the object scene. Fig. 4 depicts how topological information is embedded in the model, for a synthetic object (bottle1) and a real object from [15] (cyl1).

White kernel function. The choice of kernel function is further investigated in the experiments. For every choice, a compound function is employed, i.e. a white kernel function is added to each kernel. The white kernel function is described as

$$k^W(\mathbf{x}, \mathbf{x}'|\boldsymbol{\theta}^W = \{\sigma_W^2\}) = \sigma_W^2 \delta_{\mathbf{x},\mathbf{x}'}, \tag{10}$$

where $\delta_{\mathbf{x},\mathbf{x}'}$ is the Kronecker delta function. A fixed variance σ_W^2 is adopted, which is empirically chosen and takes value 10^{-5} for objects in the synthetic data set and 2×10^{-1}, for real sensory data set. It has been experimentally noticed that a white kernel helps to guarantee a minimal noise level in the model. As argued by [36], a white kernel function guards against overfitting, by incorporating the prior assumption that random fluctuations can happen even in very smooth underlying functions, especially since data is finite, incomplete, and possibly noisy.

Optimization procedure and constraints. The employed optimization algorithm is in the family of quasi-Newton methods. It is called L-BFGS-B [26,27], a limited-memory approximation of the Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithm that can handle simple bound constraints on variables. Its memory requirement makes it well suited for optimization problems with a large number of parameters, such as the current problem. The optimized parameters include the kernel hyperparameters, variational parameters – i.e. the inducing inputs – and the Gaussian noise variance associated to the likelihood — the latter initialized to 10^{-1} in all experiments. Two bounded constraints are defined. The inducing inputs are bounded within the object scene, i.e. the smallest centralized cube that encloses the augmented pre-processed data. Additionally the hyperparameter of the thin-plate kernel function is constrained to be at least as large as the diagonal of the object scene. This last condition is necessary to guarantee positive semi-definiteness of the kernel function. Finally, as explained in Section 3.3, the objective function to be minimized is the KL divergence from the exact joint posterior over the latent function values to the variational approximation of it, as in Eq. (5). This is equivalent to maximizing a variational lower bound of the exact log marginal likelihood.

Sampling from the posterior. Following the optimization procedure, the approximate joint posterior is calculated, as in Eq. (4). We sample query points inside the object scene, forming a $51 \times 51 \times 51$ uniform grid. We further define a grid composed by mean values of the GP posterior for every query point, which serves as input to the Marching Cubes algorithm [28], i.e. to derive the implicit surface.

4.2. Exploration and exploitation

The experiments are undertaken in two different phases: the first phase aims at exploring different model scenarios and better understanding the problem at hand, while the second phase focuses on exploiting the best possible reconstructions from the given observations.

The objects from the synthetic data set are especially useful during the first phase, since the data is considered to be clean [33] – which eases interpretation – and generally smaller. A smaller data set is an advantage in this case, because it enables full GP inference for comparative analysis. Finally the objects from the synthetic data set have more varied and challenging shape characteristics, which is especially interesting from the perspective of kernel selection.

4.2.1. Phase 1: Exploration

The exploration phase includes three experiments. The first one investigates three different kernel functions – squared exponential, Matérn and thin-plate functions – their prior assumptions and how they affect surface reconstruction. We conclude that the thin-plate kernel function best provides the structure to reconstruct various shapes. The second experiment shows how close sparse GP approximations are from full GP predictions. This is done by comparing object surfaces reconstructed through both full and sparse solutions. Finally the last experiment investigates how the variational lower bound of the exact log marginal likelihood varies with the number of inducing variables, in order to decide the number of inducing variables in the modeling scenario.

All the mentioned experiments are undertaken with Princeton ModelNet objects, particularly one object per category (apple1, bottle1, pot1 and duck1). The last experiment exceptionally includes analysis on objects from the real sensory data set as well, to verify design choices.

4.2.2. Phase 2: Exploitation

The second phase of experiments investigates the quality of object reconstructions, acquired from the robots' sensory observations, by utilizing the model design chosen in the first phase. Two experiments are included. The first experiment analyzes surface reconstructions adding tactile data and benchmarks our method against two baseline methods. We use two state-of-the-art baseline methods from the literature, which differ significantly in terms of their modeling approaches. The first baseline method [15] represents objects through a GPIS model, as does our method. In contrast, the second baseline method [39] performs shape completion by learning a general mapping, from incomplete to dense 3D point clouds, by training a deep neural network. Our method exploits the GPs ability to provide a measure of uncertainty (the variance of the GP posterior), and use this as a tool for active perception. Therefore, the first

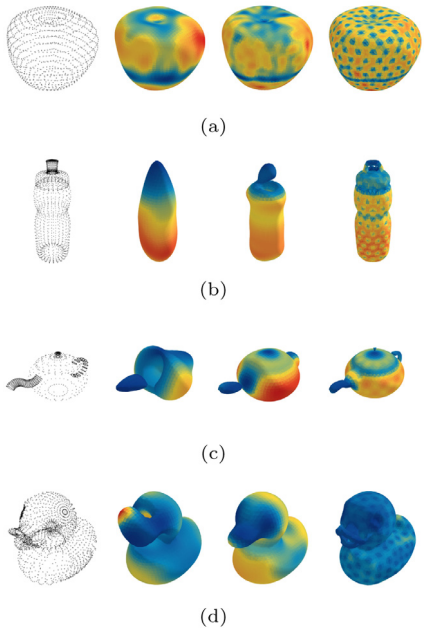


Fig. 5. Surface reconstruction for (a) apple1, (b) bottle1, (c) pot1 and (d) duck1 using different GPIS configurations. The first column displays the raw data (observations) for each object, while the second, third and fourth columns display the reconstruction results obtained with kernel functions SE, MA52 and TP, respectively.

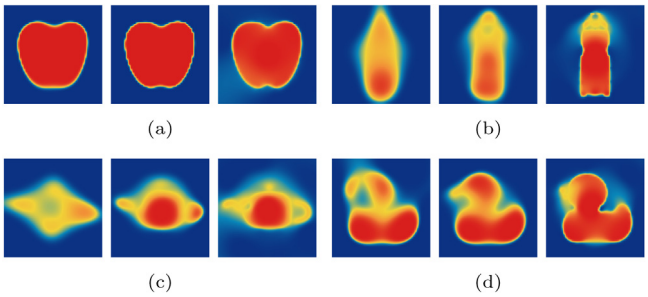


Fig. 6. Probabilistic Occupancy Maps for (a) apple1, (b) bottle1, (c) pot1 and (d) duck1 using different GPIS configurations. The first, second and third columns display the maps obtained with kernel functions SE, MA52 and TP, respectively.

experiment also includes analysis of how the surfaces evolve, when touches are strategically chosen to occur on the surface regions with highest uncertainty. The second experiment assesses reconstructions by evaluating how well objects from different categories are distinguished, based on features extracted from the reconstructed surfaces. These experiments reveal the usefulness of GPIS representations for surface reconstruction, even when very few touches are included, i.e. when data is sparse. We utilize all objects from both synthetic and real sensory data sets in these experiments.

5. Evaluation

This section introduces different evaluation tools employed to assess results.

5.1. Triangular mesh with associated uncertainty

We represent surfaces by triangular meshes. Besides providing visualizations of the 3D meshes, we visualize, for every vertex, the associated uncertainty in the estimation, based on the standard deviation from the approximate GP posterior in Eq. (4). Red color depicts surface regions where the standard deviation is the highest for the given object, i.e. regions where the surface reconstruction is the most uncertain. On the other hand, blue color depicts regions where the standard deviation is the lowest for the given object, i.e. in these regions the estimation is the most confident.

5.2. Distances to ground truth

We evaluate reconstructions by directly comparing the geometry of the reconstructed surface to the ground-truth surface. In this context, we firstly perform registration between the point cloud defined by the vertices of the mesh output by the Marching Cubes algorithm (source cloud) and the ground-truth point cloud (reference cloud). The registration is performed through the Iterative Closest Point algorithm [40]. After alignment the mean-square distance between the source and reference clouds is calculated, as well as the Hausdorff distance [41]. The last one deems two clouds as close to each other in case every point of either cloud is close to some point of the other cloud, i.e. the Hausdorff distance is defined as the greatest of all Euclidean distances from a point in one cloud to the closest point in the other cloud. These measures serve as a means to evaluate how similar the estimated surface is to the true object surface.

5.3. Probabilistic occupancy map

The Probabilistic Occupancy Map (POM) [42] is an additional evaluation tool we employ, to assess the quality of generated surfaces. POMs are useful representations of uncertainty about occupied and unoccupied space from incomplete observations.

Notice that, according to the convention adopted in Section 3.2, the probability of a point $\mathbf{x}^*$ being occupied – i.e. of belonging to the object interior – is given by

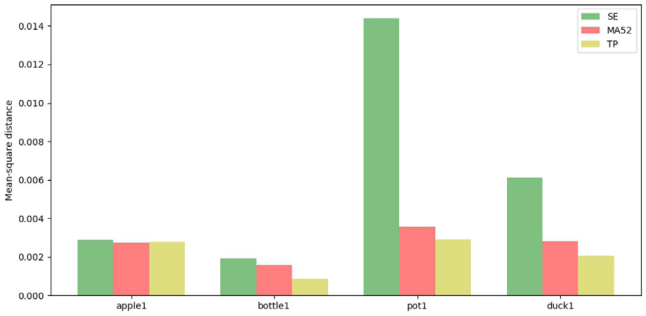
$$p(f(\mathbf{x}^*) < 0) = p(f^* < 0), \tag{11}$$

which is calculated by the cumulative distribution function of a Gaussian with mean and variance defined in Eq. (4), evaluated at zero. Note that the variance corresponds to the diagonal of the covariance matrix in the given equation.

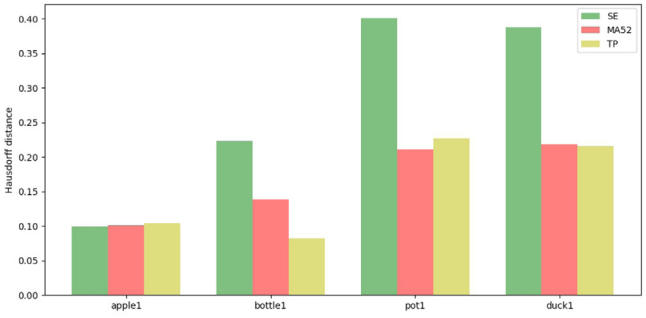
Since the input data is three-dimensional, POMs are originally maps in 3D space. For evaluation purposes, here POMs are however defined on a specific plane. The 2D POM thus displays a map with the probability of having a point occupied by the object, for every point on a defined grid. The plane is chosen to be vertical and to split the object into equal halves. We define a grid with 51×51 points. Additionally the maps depict the probabilities in terms of colors, with red meaning high occupancy probability, and blue meaning low occupancy probability. Finally the maps are slightly smoothed, for a clearer visualization.

5.4. Spectral clustering and embedding

Based on principal curvatures [29,30] from object meshes, we compute pairwise similarities between objects, as explained in Section 2.1. By relying on the top eigenvectors of a matrix derived from these pairwise similarities, spectral clustering and embedding [32] perform non-linear dimensionality reduction. For clustering the objects, we use k-means algorithm on the generated low-dimensional space, whose dimensionality in this case is equal to the number of clusters. Spectral embedding (also known as Laplacian Eigenmaps) enables visualization of clustered objects. As high-dimensional features can be very difficult to analyze, the core motivation to employ spectral embedding in



(a)



(b)

Fig. 7. The (a) mean-square and (b) Hausdorff distances for reconstructed surfaces, using the kernel functions SE, MA52 and TP.

this work is to learn – in an unsupervised fashion – a manifold of the features defined in two dimensions only. Such a representation is easily plotted, offering visual insights about the similarity between the reconstructed surfaces of analyzed objects.

6. Results

We run optimization routines until convergence, for each object.[6] Given the kernel function and all GP parameters, deriving the approximate posterior and the triangular mesh (typically required online) took in total 5.09 s, for the object with the largest point cloud (box3 in the real sensory data set from [15], with 14 572 points), and 4.76 s, for the object with the smallest point cloud (apple2, from the synthetic data set, with 832 points). Regarding the time to learn the GP parameters, it took 1 min for the smallest point cloud and 30 min for the largest point cloud. While the latter is a significant time, it is important to notice that the largest point cloud is atypically large – in comparison to the other objects – and would yield an unfeasible learning routine for full GP formulation.

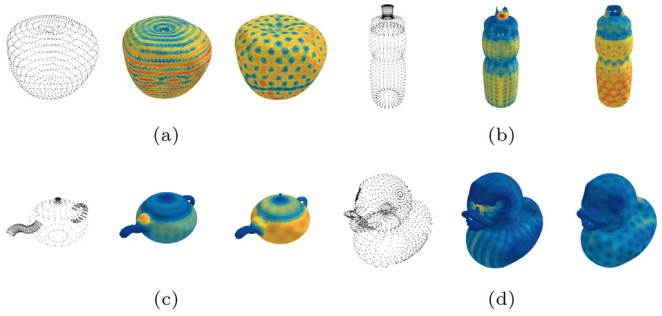
Below we address the results we obtained in each experiment described in Section 4.2.

6.1. Phase 1: Exploration

In the first experiment phase, we explore different model configurations to find best settings to be used in the next experiment phase.

6.1.1. Kernel functions

This experiment analyzed how different GPIS configurations affected the surface reconstruction. Three configurations were considered and characterized by different kernel functions: the squared exponential kernel (SE), the Matern class with $\nu = 5/2$ (MA52) and the thin-plane kernel (TP).



(a) (b) (c) (d)

Fig. 8. Surface reconstruction for (a) apple1, (b) bottle1, (c) pot1 and (d) duck1 using GPIS with full and sparse GPs. The first column displays the raw data (observations) for each object, while the second and third columns display the reconstruction results obtained with full and sparse GPs, respectively.

Fig. 5 displays the reconstructed surfaces, with associated predictive uncertainties, while the associated POMs can be seen in Fig. 6. The different configurations were also analyzed in terms of distances between point clouds. Fig. 7 displays two graphs with the mean-square and Hausdorff distances between vertices of the triangular mesh and the ground-truth point cloud, for the analyzed kernel functions.

For every choice of kernel function, a compound function was employed, i.e. a white kernel function was added to each kernel. We employ $\sigma_W^2 = 10^{-5}$ for the GPIS configuration with TP. For configurations with SE and MA52, we explore different values for σ_W^2 – 10^{-5}, 10^{-4}, 5×10^{-3}, 10^{-3}, 10^{-2} and 10^{-1} – and only the best reconstruction result (based on distance measures and visual inspection) for each object is considered. Additionally, the kernel hyperparameters were initialized in the following way: the SE and MA52 variances were initialized as 0.5, while the lengthscales, as 2.3. The TP hyperparameter R was initialized with the length of the diagonal of the object scene.

As Figs. 5, 6 and 7 show, in general the estimated object surfaces vary considerably for different kernel choices. The SE was not able to satisfactorily capture object surfaces with sharp-edged contours, producing smooth surfaces, since this kernel function is infinitely differentiable and therefore tends to model well the smoothness characteristics of random processes. The apple is the only object that was well reconstructed by SE, since it has originally a very smooth surface.

The MA52 presented better results, as it was able to output better surface reconstructions, capturing more surface details. Note for instance how the POM for the pot is significantly more reliable for MA52 than for SE. We believe that, since MA52 is more generalized and less differentiable (two-times differentiable), it is able to model a more varied set of shape characteristics.

GPIS with TP in turn demonstrated to capture shape information in the most reliable way, among the 3 analyzed GPIS configurations. This can be noticed in Fig. 5: a great amount of surface details is captured, such as sharp-edged contours. Fig. 6 displays more confident POMs for TP, i.e. the TP reconstructions are more accurate in terms of occupancy probabilities as well. Furthermore, Fig. 7 suggests that the GPIS with TP generates reconstructions that are in general closer to the ground-truth object, in comparison to SE and MA52. Given the positive results achieved with TP, the GPIS configurations employed in the remaining of this work use kernel function TP.

Finally it can be noticed that some details – such as the bottle cap, pot handle, pot pipe, pot lid's nob and duck's beak – may

[6] All computational operations, including optimization of parameters and sparse GP inference, were carried out on a PC with 8 GB of RAM memory and processor Intel(R) Core(TM) i7-3520M CPU @ 2.90 GHz with 4 cores.

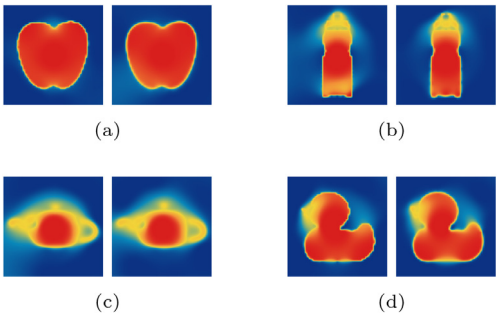


Fig. 9. Probabilistic Occupancy Maps for (a) apple1, (b) bottle1, (c) pot1 and (d) duck1 using GPIS with full GP (left) and sparse GP (right).

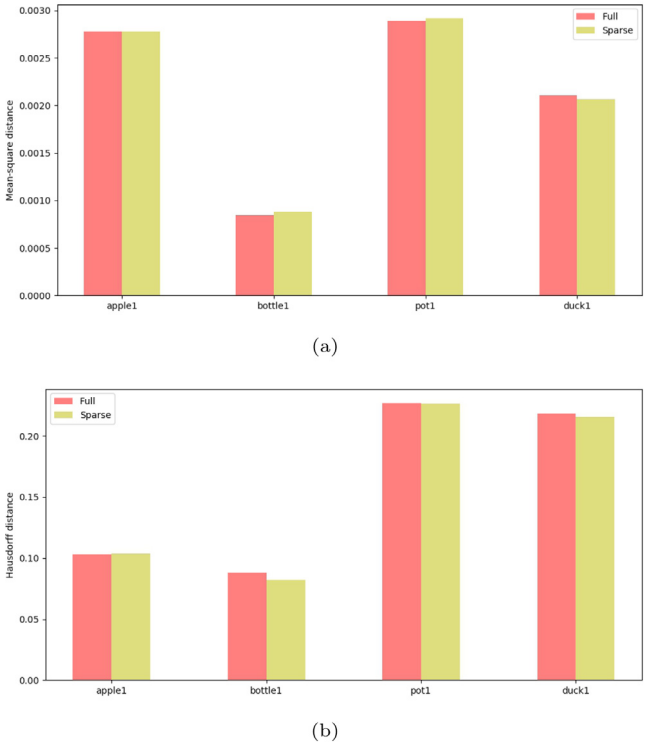


Fig. 10. The (a) mean-square and (b) Hausdorff distances for full and sparse GP reconstructions.

not be well represented, even by GPIS with TP. Fig. 6 displays this phenomenon, as POMs decrease the occupancy probability on these regions. We believe that the topological information provided to the model plays an important role and we plan to improve the way we include topological information (Section 4.1) as a future work.

6.1.2. Sparse and full Gaussian processes

This experiment was performed as a means to compare sparse and full GPIS formulations. We remind the reader that the sparse GP we employ has a variational formulation, which implies that optimization of the GP parameters attempts to approximate the full GP solution directly.

Figs. 8 and 9 display the reconstructed surfaces and the POMs, respectively. It can be noticed from Fig. 8 that surface reconstructions with full GPs are more confident, i.e. the predictive standard deviation is lower. This is expected, since the sparse representation is an approximation to the full representation and therefore should be more uncertain. The POMs in both configurations in Fig. 9 look very similar. However, a closer look reveals

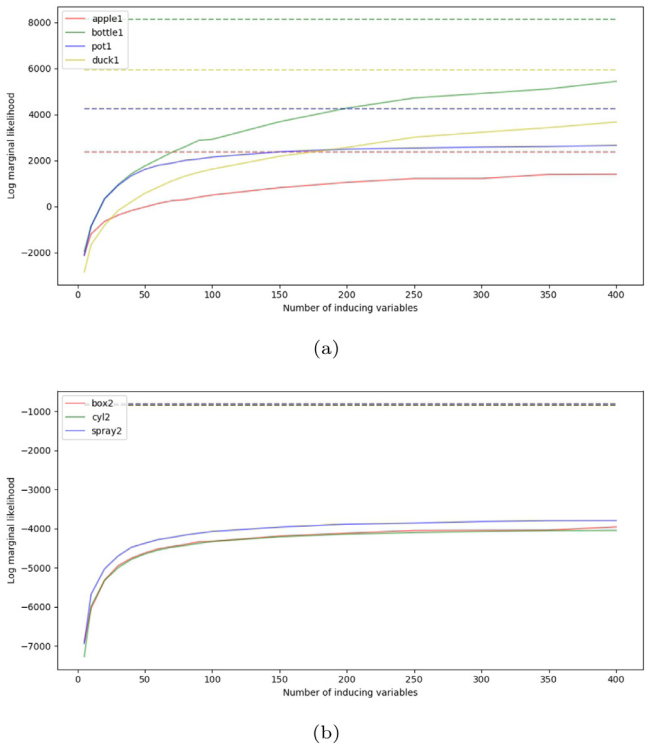


Fig. 11. The variational lower bound of the exact log marginal likelihood, for different numbers of inducing variables, using the (a) synthetic data set and (b) real sensory data set from [15].

that the POMs for sparse solutions display less precise contours, as if the borders of the objects are smoother. This relates to the previous comment, i.e. by inducing sparsity one generates a more uncertain probabilistic representation. Nevertheless, the mean-square and Hausdorff distances between vertices of the triangular mesh and the ground-truth point cloud, in Fig. 10, show that estimations given by full and sparse solutions are very similar, which confirms that the sparse GP well approximates its full counterpart. We continue employing sparse GPs throughout the remaining experiments.

Finally it was noticed that the learned variance of the Gaussian noise was generally very low (smaller than 10^{-7}) for full GP solutions. We believe that this explains some surface artifacts given by full GPs, e.g. as on the reconstructed duck (especially on the duck's face). We observed that this phenomenon can be corrected by setting a higher variance for the white kernel σ_W^2.

6.1.3. Number of inducing variables

Through this experiment we could observe how the variational lower bound of the exact log marginal likelihood evolved with the number of inducing variables, for objects in the synthetic data set and real sensory data set from [15]. From the later data set we use objects box2, cyl2 and spray2.[7] Fig. 11 summarizes the findings. Solid lines represent the variational lower bounds, while dashed lines represent the exact solutions, obtained by full GP inference. The bound gets tighter as the number of inducing variables increases, even though the gaps are different for different objects.

We agree that, as long as the bound decreases, our reconstructions are better. However, we noticed that this fact does not aid us

[7] The data from objects box2, cyl2 and spray2 are small enough to enable full GP inference.

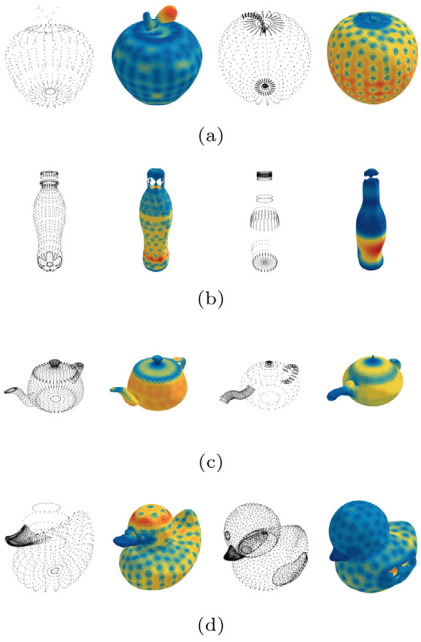


Fig. 12. Surface reconstruction for (a) apple2 and apple3, (b) bottle2 and bottle3, (c) pot2 and pot3 and (d) duck2 and duck3. The first and third columns display the raw data (observations) for each object, while the second and fourth columns display the reconstruction results for the respective objects.

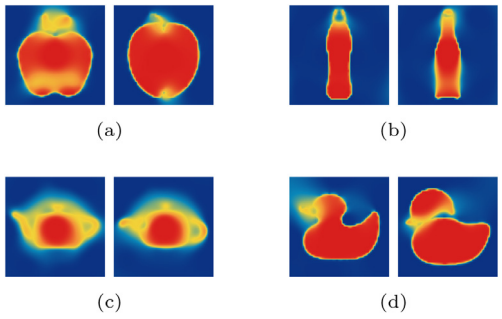


Fig. 13. Probabilistic Occupancy Maps for (a) apple2 and apple3, (b) bottle2 and bottle3, (c) pot2 and pot3 and (d) duck2 and duck3.

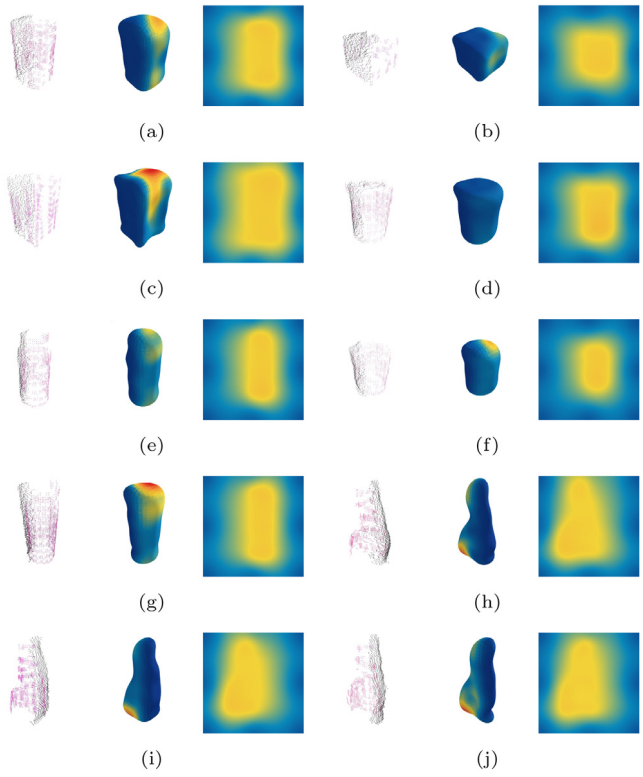


Fig. 14. Surface reconstruction and POMs for (a) box1, (b) box2, (c) box3, (d) cyl1, (e) cyl2, (f) cyl3, (g) cyl4, (h) spray1, (i) spray2 and (j) spray3, objects in the real sensory data set from [15]. The first column displays the raw data (observations, where black and purple colors stand for visual and tactile data points, respectively) for each object, the second column displays the reconstruction results and the third column, the POMs.

to choose an appropriate number of inducing variables, because we can only perceive the improvements on a relative scale, not on an absolute scale. This is a general limitation of variational methods.

Given this limitation and in order to choose the number of inducing variables to our problem, we performed experiments with 250, 350, 450 and 750 inducing variables. The time it takes to learn GP parameters and derive the approximate posterior increases with the number of inducing variables. We noticed that, compared to setting 350 inducing variables, setting 450 and 750 inducing variables increases the time to learn GP parameters by more than 50% and more than 100%, respectively, averaging over objects box2, cyl2 and spray2. Therefore, for the evaluations in this work, we set the number of inducing variables to 350, which enables good reconstruction results without offering a significant computational burden (as shown by the computational times reported in the beginning of Section 6). Finally, we decided to initialize the inducing inputs by sampling surface points and taking all external and internal points as well (as explained in Section 4.1).

6.2. Phase 2: Exploitation

Based on the insights collected in the first phase of experiments, we expand our analysis to a wider range of objects in terms of surface reconstructions and object clustering.

6.2.1. Surface reconstruction

Firstly we analyzed the outcomes of surface reconstruction for different objects. These findings were derived for all evaluated data sets and are addressed below.

Princeton modelnet synthetic data set. Fig. 12 summarizes how the remaining objects in the synthetic data set are reconstructed. The generated triangular meshes resemble the original objects. However we notice artifacts on some surface representations, particularly on regions with details, such as bottle caps, as illustrated in Fig. 13, which shows that the occupancy probabilities are usually not high on such regions. In particular, the raw data from bottle3 is incomplete in some regions. Note in Fig. 12b how the input points are scarce on the side of the bottle, with virtually no data point on a large area. Therefore, the standard deviation is high on the surface region where data is scarce. Additionally, the POM for bottle3 in Fig. 13b presents uncertain edges, justified by the lack of data points, in comparison to the POM for bottle2, which has a more complete point cloud.

Real sensory data set. Fig. 14 displays how the objects in the real sensory data set from [15] are reconstructed, for boxes, cylinders and spray bottles. Note that, in comparison to the objects in the synthetic data set, here the estimations are coarser. This is justified by the great amount of noise in the data. Notice additionally

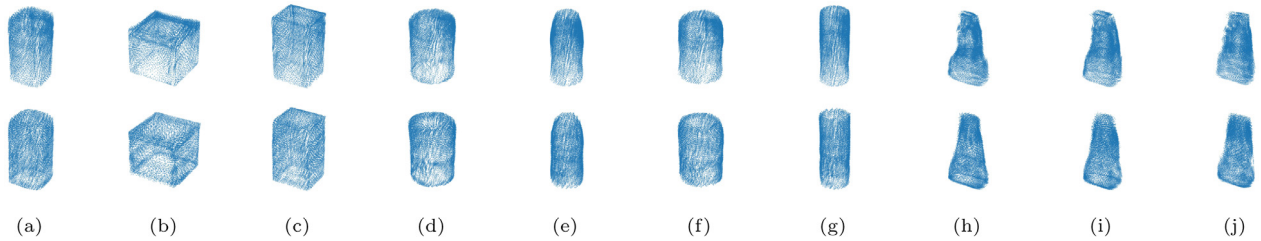


Fig. 15. Shape completion as proposed by [39], for (a) box1, (b) box2, (c) box3, (d) cyl1, (e) cyl2, (f) cyl3, (g) cyl4, (h) spray1, (i) spray2 and (j) spray3 , objects in the real sensory data set from [15]. The top and bottom rows are derived from the pre-trained models PCN-CD and PCN-EMD, respectively.

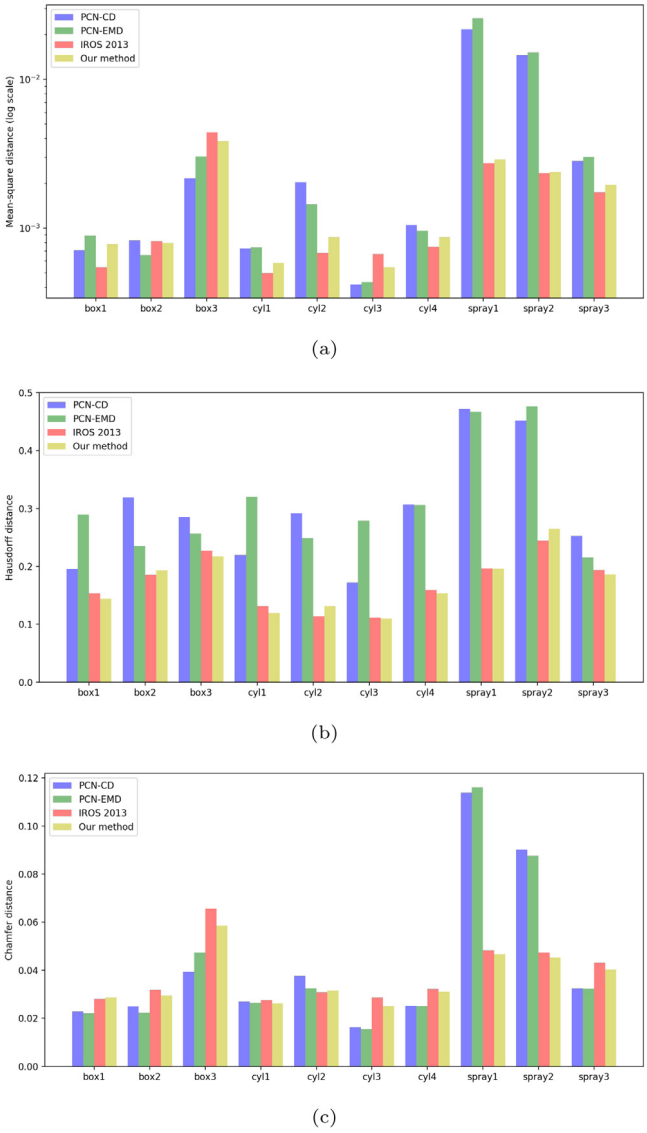


Fig. 16. The (a) mean-square, (b) Hausdorff and (c) Chamfer distances for shape completion as proposed by [39] and for reconstructions undertaken according to the method in [15] and our method.

that the POMs are less certain about where the object contours should lie.

We compare our approach to two baseline methods, [15] and [39]. The latter uses encoder–decoder deep neural networks to perform shape completion on point clouds in a coarse-to-fine fashion, by training the networks on pairs of partial and complete point clouds derived from the ShapeNet [43] data set. To train these networks, more than 30 000 objects from 8 different categories (airplane, cabinet, car, chair, lamp, sofa, table and vessel) were used. To calculate a loss function that operates on point clouds, [39] employs two permutation-invariant functions for point sets, introduced by [44]: the Chamfer and Earth Mover's distances. In particular, the loss function is the sum of two separate terms that measure quality of coarse and fine outputs. Two pre-trained networks are introduced, namely PCN-CD and PCN-EMD. While both employ the Chamfer distance to assess the fine output, PCN-CD and PCN-EMD differ on the choice of function for the coarse output assessment: Chamfer distance for PCN-CD and Earth Mover's distance for PCN-EMD. We fed the visual and tactile points of objects in the real sensory data set from [15] to the two pre-trained networks of [39], and show the resulting point clouds in Fig. 15, which contain 16 384 points each.

We computed the mean-square, Hausdorff and Chamfer distances between vertices in the output triangular meshes (for [15] and our method), the output complete point clouds (for [39]) and the ground-truth point clouds. We include the symmetric Chamfer distance in this analysis, as it is used in [39]. Fig. 16 displays the results. In comparison to [39], our method always yields smaller Hausdorff distances. In terms of mean-square distance, our method gives often better results than [39] for sprays and cylinders, but not for boxes. Based on Chamfer distance, our method leads to better results mostly for spray bottles. Taking a closer look, note that our method in general performs better among objects with few symmetry planes and more irregular shapes, such as spray bottles, while [39] is better at completing the shape of most objects which have multiple symmetry planes. This suggests that the neural network appears to have implicitly learned to make symmetry assumptions based on the used training set. Note that, despite offering good shape completion for some objects, e.g. the sharp-edged boxes, a significant limitation of [39] is that the methods do not directly generate a continuous watertight surface, and do not describe topology. Additionally, our approach offers a useful feature for active exploration of unknown surfaces, based on uncertainty being encoded in our estimated shape models. The distance metrics for [15] and our method are mostly similar. This indicates that the sparse GP formulation of our method (in contrast to the full GP of [15]) with optimized parameters (in contrast to empirically selected parameters of [15]) yields sufficient approximations and competitive quality in surface reconstruction.

Additionally, we compared the computational times to derive the posterior and the triangular mesh, using the sparse and full GP formulations, for all objects in the real sensory data set from [15]. We observed that for sparse GPs the computational times are at least 4 times shorter than for full GPs. Finally, we perform surface reconstruction on additional objects whose data was collected using a PR2 robot. Fig. 17 shows the results for boxes, cylinders and bottles.

Besides the presented reconstruction results using full tactile data, we analyzed how the surfaces evolve when touches are dynamically selected on the surface area for which information

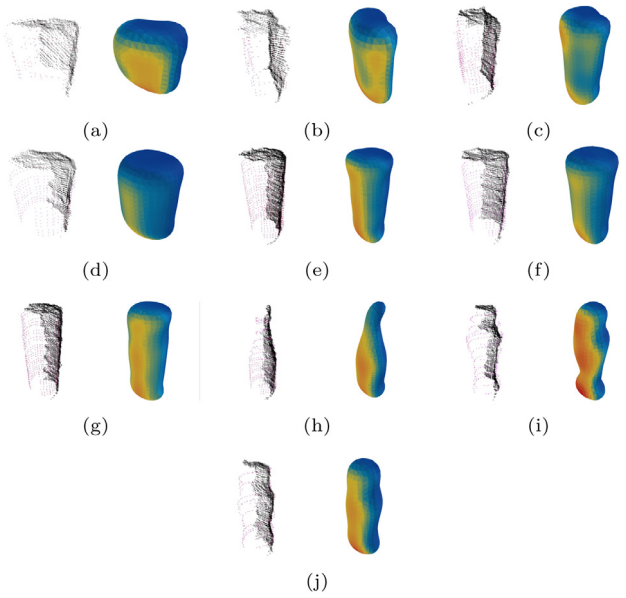


Fig. 17. Surface reconstruction for (a) box1, (b) box2, (c) box3, (d) cyl1, (e) cyl2, (f) cyl3, (g) cyl4, (h) bottle1, (i) bottle2 and (j) bottle3, objects in the real sensory data set collected using a PR2 robot. For each object, the raw data (observations, where black and purple colors stand for visual and tactile data points, respectively) is displayed on the left, while the reconstruction results are displayed on the right.

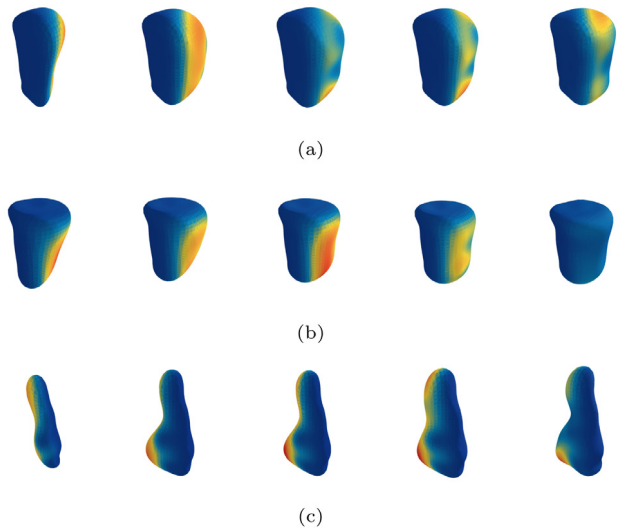


Fig. 18. Surface reconstructions for (a) box1, (b) cyl1 and (c) spray1, created with increasing number of touch actions, which were selected based on the surface area with maximum variance of GP posterior. From left to right: 0, 1, 2, 3 and finally all touches are considered.

gain is maximum. GPs provide an inherent measure of prediction uncertainty (the variance of the GP posterior), which we use to increase information gain for every selected touch. The real sensory data set from [15] is used to perform this analysis. We start by calculating the reconstructed surface for every object, using its visual data points only. Then the best action to be taken next is selected from a discrete action space defined by the vertical position (6 different heights) and the approaching angle (9 different angles). The vertical position and the approaching angle are computed with respect to the centroid of the current model. For each possible action, the closest point to the tactile sensor pads is calculated on the triangular mesh. Finally, the action selected is the one for which the GP posterior variance is maximum, for all tactile pads. After acquiring the tactile readings, the surface is updated and the best next action is repeatedly identified through the same routine.

Fig. 18 displays the reconstructed surfaces of objects box1, cyl1 and spray1, when only visual data is used, after the first, second and third touch actions are performed and, finally, after full tactile data is acquired. Note that the surfaces change drastically already when few touch actions are performed. Particularly note that the surfaces after 3 touches get roughly similar to the final surfaces, despite using less tactile readings. This shows that our method for surface reconstruction offers quick convergence of triangular meshes, when touch actions are optimally chosen to increase information gain.

6.2.2. Object clustering

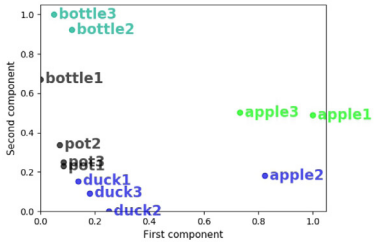
Spectral clustering was based on pairwise similarities between sets of principal curvatures calculated from object meshes. The lower-dimensional representations for each object were fed to the k-means algorithm, which was run 100 times with different initializations. The final result corresponds to the best output of all runs in terms of inertia (or within-cluster sum-of-squares).

The spectral clustering and embedding can be seen in Fig. 19. Notice how the 2D embeddings separate the different categories in the 2D space. Particularly in Fig. 19a the spectral clustering discriminates the objects correctly for 10 out of 12 objects (apple2 and bottle1 are incorrectly classified). Notice that, if the number of clusters is decreased to $t = 3$ (Fig. 19b), the pots and ducks are agglutinated in one single cluster. In this case, it can be said that the cluster with apples represents round objects, the cluster with bottles represents general cylindrical objects, while the one with pots and ducks have a more diverse set of curvatures.

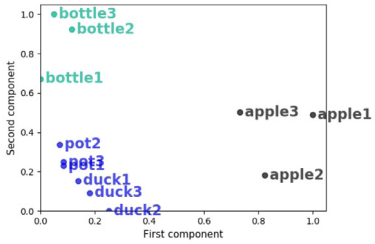
The spectral clustering and embedding for real sensory data can be seen in Figs. 19c, 19d and 19e. Particularly Fig. 19c displays clustering and embedding performance after the third touch action was performed, for every object in the real sensory data set from [15]. Note that, even though only 3 touches were performed, all objects can already be distinguished correctly, since information gain is maximized for every touch. This confirms that convergence of the principal curvatures for the proposed method is fast, enabling efficient active tactile exploration for perception of shape. For comparison, Fig. 19d displays the spectral clustering and embedding using full tactile data for every object. Finally Fig. 19e reflects the same analysis on the real sensory data set from the PR2 robot, using full tactile data. On the whole, it can be concluded that the significant presence of noise in the real sensory data (especially in the data collected using a PR2 robot) did not prevent the clustering algorithm from distinguishing the objects from different categories. This is hence a confirmation that the representation from noisy data, obtained through GPIS based on sparse GP with variational formulation, preserved meaningful shape attributes.

7. Conclusion

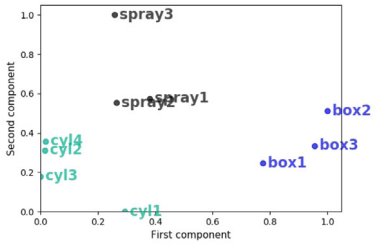
We presented a comprehensive evaluation of probabilistic representations to capture shape information from visual and tactile data. Unlike deterministic models for surface reconstruction, our models imply a distribution over surfaces and parameterize the predictive uncertainty, which provides crucial information about object shape. We evaluated our approach by comparing the estimated surfaces with the ground-truth surfaces and by clustering objects based on acquired models. A comparative analysis with the results from [15] was also provided, which showed that the proposed approach does not reduce the quality of resulting models, even though we adopt an approximate method with induced



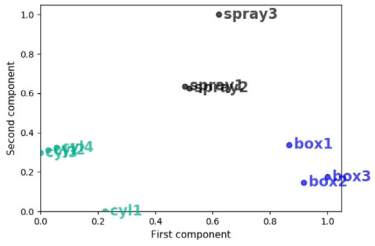
(a)



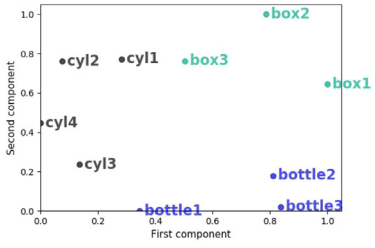
(b)



(c)



(d)



(e)

Fig. 19. Spectral embedding in 2D with spectral clustering in colors, for objects in the (a, b) synthetic data set, real sensory data set (c, d) from [15] and (e) from a PR2 robot. The number of clusters is set as (a) four clusters and (b, c, d, e) three clusters. In particular, the embedding and clustering (c) after 3 touches is compared to the ones (d) after exhaustive tactile exploration. Note from (c, d) that optimal clustering performance is achieved after 3 touches already. The axes correspond to the first and second components of the 2D representation [32].

sparsity. Additionally, we benchmarked our method against a state-of-the-art baseline method [39]. We showed that our approach provides comparable results in general. However, our approach generates better reconstructions, more closely resembling ground truth, for most objects that are less symmetrical and more irregular in shape. Finally, we showed that our surface estimates converge fast, demonstrated by analyzing surfaces and clustering performance. This fast convergence is due to: (i) selecting touch actions for efficient haptic exploration, to maximize information gain, based on the variance of the GP posterior; (ii) our use of a sparse GPIS formulation. Hence, by optimizing information gain for every touch action, we provide a framework that minimizes the number of touches necessary to extract useful shape information about objects.

We plan to extend our work to use acquired models for grasp and manipulation planning, to improve the framework by extending it to learn models per shape categories and by enhancing the data with surface normals, that provide further topological information. We also plan to use heteroscedastic noise in our modeling, which will enable us to have different noise levels for different modalities. Another interesting future direction is to use geometric priors and other kernel functions, such as non-stationary kernels, which can help the model adapt better to surfaces whose smoothness varies for different locations.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was partly supported by: EU CHIST-ERA project on Perception-Guided Robotic Grasping; EPSRC projects EP/P017487/1 and EP/R02572X/1 (National Center for Nuclear Robotics), and the Faraday Institution ReLiB project FIRG005/FIRG006. Prof. Rustam Stolkin was sponsored by a Royal Society Industry Fellowship.

References

[1] N. Gaissert, C. Wallraven, Integrating visual and haptic shape information to form a multimodal perceptual space, in: IEEE World Haptics Conference (WHC), 2011, pp. 451–456.

[2] M.O. Ernst, M.S. Banks, Humans integrate visual and haptic information in a statistically optimal fashion, Nature 415 (2002) 429–433.

[3] H.B. Helbig, M.O. Ernst, Optimal integration of shape information from vision and touch, Exp. Brain Res. 179 (2007) 595–606.

[4] N. Vahrenkamp, M. Do, T. Asfour, R. Dillmann, Integrated grasp and motion planning, in: 2010 IEEE International Conference on Robotics and Automation, 2010, pp. 2883–2888.

[5] Z.J. Yifan Hou, M.T. Mason, Fast planning for 3d any-pose-reorienting using pivoting, in: International Conference on Robotics and Automation (ICRA) 2018, IEEE Robotics and Automation Society (RAS), 2018.

[6] K. Pauwels, D. Kragic, SimTrack: A simulation-based framework for scalable real-time object pose detection and tracking, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, Hamburg, Germany, 2015, pp. 1300–1307, http://dx.doi.org/10.1109/IROS.2015.7353536.

[7] D. Schiebener, A. Schmidt, N. Vahrenkamp, T. Asfour, Heuristic 3d object shape completion based on symmetry and scene context, in: 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2016, pp. 74–81, http://dx.doi.org/10.1109/IROS.2016.7759037.

[8] L. Sun, C. Zhao, Z. Yan, P. Liu, T. Duckett, R. Stolkin, A novel weakly-supervised approach for rgb-d-based nuclear waste object detection, IEEE Sens. J. 19 (9) (2019) 3487–3500, http://dx.doi.org/10.1109/JSEN.2018.2888815.

[9] G. Harper, R. Sommerville, E. Kendrick, L. Driscoll, P. Slater, R. Stolkin, A. Walton, P. Christensen, O. Heidrich, S. Lambert, A. Abbott, K. Ryder, L. Gaines, P. Anderson, Recycling lithium-ion batteries from electric vehicles, Nature 575 (7781) (2019) 75–86, http://dx.doi.org/10.1038/s41586-019-1682-5.

[10] N. Marturi, A. Rastegarpanah, V. Rajasekaran, V. Ortenzi, Y. Bekiroglu, J. Kuo, R. Stolkin, Towards advanced robotic manipulations for nuclear decommissioning, in: H. Canbolat (Ed.), Robots Operating in Hazardous Environments, IntechOpen, Rijeka, 2017, http://dx.doi.org/10.5772/intechopen.69739.

[11] M. Krainin, B. Curless, D. Fox, Autonomous generation of complete 3d object models using next best view manipulation planning, in: 2011 IEEE International Conference on Robotics and Automation, 2011, pp. 5031–5037, http://dx.doi.org/10.1109/ICRA.2011.5980429.

[12] J. Ilonen, J. Bohg, V. Kyrki, Fusing visual and tactile sensing for 3-d object reconstruction while grasping, in: IEEE International Conference on Robotics and Automation (ICRA), 2013, pp. 3547–3554.

[13] S. Dragiev, M. Toussaint, M. Gienger, GaussIan process implicit surfaces for shape estimation and grasping, in: 2011 IEEE International Conference on Robotics and Automation, 2011, pp. 2845–2850, http://dx.doi.org/10.1109/ICRA.2011.5980395.

[14] J. Mahler, S. Patil, B. Kehoe, J.P. van den Berg, M.T. Ciocarlie, P. Abbeel, K.Y. Goldberg, Gp-gpis-opt: Grasp planning with shape uncertainty using Gaussian process implicit surfaces and sequential convex programming, in: 2015 IEEE International Conference on Robotics and Automation (ICRA), 2015, pp. 4919–4926.

[15] M. Björkman, Y. Bekiroglu, V. Högman, D. Kragic, Enhancing visual perception of shape through tactile glances, in: 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2013, pp. 3180–3186, http://dx.doi.org/10.1109/IROS.2013.6696808.

[16] S. Caccamo, Y. Bekiroglu, C.H. Ek, D. Kragic, Active exploration using Gaussian random fields and Gaussian process implicit surfaces, in: 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2016, pp. 582–589, http://dx.doi.org/10.1109/IROS.2016.7759112.

[17] Z. Yi, R. Calandra, F. Veiga, H. van Hoof, T. Hermans, Y. Zhang, J. Peters, Active tactile object exploration with gaussian processes, in: 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2016, pp. 4925–4930, http://dx.doi.org/10.1109/IROS.2016.7759723.

[18] D. Driess, P. Englert, M. Toussaint, Active learning with query paths for tactile object shape exploration, in: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2017, pp. 65–72, http://dx.doi.org/10.1109/IROS.2017.8202139.

[19] N. Jamali, C. Ciliberto, L. Rosasco, L. Natale, Active perception: Building objects' models using tactile exploration, in: 2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids), 2016, pp. 179–185, http://dx.doi.org/10.1109/HUMANOIDS.2016.7803275.

[20] T. Matsubara, K. Shibata, K. Sugimoto, Active touch point selection with travel cost in tactile exploration for fast shape estimation of unknown objects, in: 2016 IEEE International Conference on Advanced Intelligent Mechatronics (AIM), 2016, pp. 1115–1120, http://dx.doi.org/10.1109/AIM.2016.7576919.

[21] O. Williams, A. Fitzgibbon, GaussIan process implicit surfaces, in: Gaussian Processes in Practice, 2006, URL https://www.microsoft.com/en-us/research/publication/gaussian-process-implicit-surfaces/.

[22] W. Martens, Y. Poffet, P.R. Soria, R. Fitch, S. Sukkarieh, Geometric priors for gaussian process implicit surfaces, IEEE Robot. Autom. Lett. 2 (2) (2017) 373–380, http://dx.doi.org/10.1109/LRA.2016.2631260.

[23] R. Hadsell, J.A. Bagnell, D. Huber, M. Hebert, Space-carving kernels for accurate rough terrain estimation, Int. J. Robot. Res. 29 (8) (2010) 981–996, http://dx.doi.org/10.1177/0278364910369996, arXiv:http://dx.doi.org/10.1177/0278364910369996.

[24] C.E. Rasmussen, C.K.I. Williams, Gaussian Processes for Machine Learning, in: (Adaptive Computation and Machine Learning), The MIT Press, 2006.

[25] M.K. Titsias, Variational learning of inducing variables in sparse Gaussian processes, in: D.V. Dyk, M. Welling (Eds.), Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics (AISTATS-09), Vol. 5, Journal of Machine Learning Research - Proceedings Track, 2009, pp. 567–574, URL http://jmlr.csail.mit.edu/proceedings/papers/v5/titsias09a/titsias09a.pdf.

[26] R.H. Byrd, P. Lu, J. Nocedal, C. Zhu, A limited memory algorithm for bound constrained optimization, SIAM J. Sci. Comput. 16 (5) (1995) 1190–1208, http://dx.doi.org/10.1137/0916069.

[27] C. Zhu, R.H. Byrd, P. Lu, J. Nocedal, Algorithm 778: L-bfgs-b: Fortran subroutines for large-scale bound-constrained optimization, ACM Trans. Math. Software 23 (4) (1997) 550–560, http://dx.doi.org/10.1145/279232.279236, URL http://doi.acm.org/10.1145/279232.279236.

[28] T.M. Lewiner, H.C.V. Lopes, A.W. Vieira, G.T.d. Santos, Efficient implementation of marching cubes' cases with topological guarantees, J. Graph. Tools 8 (2) (2003) 1–15.

[29] X. Chen, F. Schmitt, Intrinsic surface properties from surface triangulation, in: Computer Vision — ECCV'92: Second European Conference on Computer Vision Santa Margherita Ligure, Italy, May 19–22, 1992 Proceedings, Vol. 588, 1992, pp. 739–743. http://dx.doi.org/10.1007/3-540-55426-2_83.

[30] C.-s. Dong, G.-z. Wang, Curvatures estimation on triangular mesh, J. Zhejiang Univ.-Sci. A 6 (1) (2005) 128–136, http://dx.doi.org/10.1007/BF02887228.

[31] A. Gretton, K.M. Borgwardt, M.J. Rasch, B. Schölkopf, A. Smola, A kernel two-sample test, J. Mach. Learn. Res. 13 (2012) 723–773, URL http://dl.acm.org/citation.cfm?id=2188385.2188410.

[32] A.Y. Ng, M.I. Jordan, Y. Weiss, On spectral clustering: Analysis and an algorithm, in: T.G. Dietterich, S. Becker, Z. Ghahramani (Eds.), Advances in Neural Information Processing Systems, Vol. 14, MIT Press, 2002, pp. 849–856, URL http://papers.nips.cc/paper/2092-on-spectral-clustering-analysis-and-an-algorithm.pdf.

[33] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, J. Xiao, 3D ShapeNets: A deep representation for volumetric shape modeling, in: Proceedings of 28th IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1912–1920, URL http://arxiv.org/abs/1406.5670.

[34] Princeton modelnet, 2014, http://modelnet.cs.princeton.edu/ (Accessed: 2019-05-20).

[35] J. Quiñonero Candela, C.E. Rasmussen, A unifying view of sparse approximate gaussian process regression, J. Mach. Learn. Res. 6 (2005) 1939–1959, URL http://dl.acm.org/citation.cfm?id=1046920.1194909.

[36] A. Damianou, Deep Gaussian Processes and Variational Propagation of Uncertainty (Ph.D. thesis), University of Sheffield, 2015.

[37] M.G. Genton, Classes of kernels for machine learning: A statistics perspective, J. Mach. Learn. Res. 2 (2001) 299–312, URL http://www.jmlr.org/papers/v2/genton01a.html.

[38] C.M. Bishop, Pattern Recognition and Machine Learning, Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[39] W. Yuan, T. Khot, D. Held, C. Mertz, M. Hebert, PCN: Point completion network, in: 2018 International Conference on 3D Vision (3DV), 2018, pp. 728–737.

[40] P.J. Besl, N.D. McKay, A method for registration of 3-d shapes, IEEE Trans. Pattern Anal. Mach. Intell. 14 (2) (1992) 239–256, http://dx.doi.org/10.1109/34.121791.

[41] F. Hausdorff, Grundzüge der Mengenlehre, Veit and Company, Leipzig, 1914, Das Hauptwerk von Felix Hausdorff. URL https://archive.org/details/grundzgedermen00hausuoft.

[42] F. Fleuret, J. Berclaz, R. Lengagne, P. Fua, Multicamera people tracking with a probabilistic occupancy map, IEEE Trans. Pattern Anal. Mach. Intell. 30 (2) (2008) 267–282, http://dx.doi.org/10.1109/TPAMI.2007.1174.

[43] A.X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, F. Yu, Shapenet: An information-rich 3d model repository, 2015, cite arxiv:1512.03012 URL http://arxiv.org/abs/1512.03012.

[44] H. Fan, H. Su, L.J. Guibas, A point set generation network for 3d object reconstruction from a single image, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017, 2017, pp. 2463–2471, http://dx.doi.org/10.1109/CVPR.2017.264.

Gabriela Zarzar Gandler received her Master of Science degree from the Royal Institute of Technology (KTH) in 2017. She is currently working as AI Research Engineer at Peltarion, in Stockholm, Sweden. Prior to Peltarion she worked at ABB Corporate Research Center and at Looklet AB.

Carl Henrik Ek is a Senior Lecturer in Computer Science at the University of Bristol, UK and a Docent in Machine Learning at the Royal Institute of Technology, Sweden. Before joining Bristol he was an Assistant Professor in Machine Learning at the Royal Institute of Technology (KTH) in Stockholm. He did his postdoctoral research at University of California at Berkeley. His PhD is from Oxford Brookes University.

Mårten Björkman is an Associate Professor and director of studies in Computer Science at KTH, Sweden. He received a PhD in computer vision at KTH in 2002. His primary research interests are stereo vision, cognitive vision systems and image-based rendering.

Prof. Rustam Stolkin was awarded an M.Eng. in Engineering from Oxford University, 1998, and a PhD in Robotic Vision, 2004, undertaken between University College London and industry. He is concurrently: founder and Director of UK's National Center for Nuclear Robotics; Royal Society Industry Fellow; Chair of Robotics at University of Birmingham; Director of A.R.M Robotics Ltd, which has done pioneering work for the nuclear industry. Professor Stolkin is highly interdisciplinary, with patents and publications spanning: vision and imaging; learning and AI; grasping and manipulation; robot vehicles; human–robot interaction; and extensive work on science and engineering education.

Yasemin Bekiroglu completed her Ph.D. at the Royal Institute of Technology (KTH), Sweden, in 2012. She worked as a research scientist at ABB Corporate Research Center, Sweden, as a roboticist at Vicarious, California, and as a post-doctoral researcher at University of Birmingham. Her research is focused on data driven learning for robotics applications with a focus on Bayesian non-parametrics. In specific she is interested in data efficient learning from multisensory data.